

一种贪心策略的更高效的请求集生成算法*

李美安, 陈志党, 王春申

(内蒙古农业大学 计算机科学与技术学院, 内蒙古 呼和浩特 010018)

摘要: 在折半循环编码算法的基础上, 依据贪心策略对可纳入节点进行局部求最优的方式来生成请求集的算法, 从而使算法的请求集长度下降了一个数量级, 接近 \sqrt{N} 。

关键词: 初始化; 折半循环编码; 局部贪心策略; 请求集

中图分类号: TP393

文献标识码: A

文章编号: 1674-7720(2011)13-0052-03

A greedy strategy for more efficient request set generation algorithm

Li Meian, Chen Zhidang, Wang Chunshen

(College of Computer and Information Engineering, Inner Mongolia Agricultural University, Huhhot 010018, China)

Abstract: Based on the Bin-cyclic, this paper generates the request set of algorithm by getting the local optimization solution of the demand nodes through greedy strategy. Thereby, the algorithm's the length of the request set drops an order of magnitude, which close to \sqrt{N} .

Key words: initialization; Bin-cyclic; local greedy strategy; the request set

基于请求集的分布式互斥算法作为 Maekawa 算法^[1]的推广, 近年来得到了人们的广泛关注, 人们提出了许多各具特色的算法来构建请求集, 以降低分布式互斥算法的消息复杂度或者提高分布式互斥算法在其他方面的性能, 例如李美安通过循环编码产生请求集的方式, 得出一种消息复杂度较低、容错性能高且同步时间短的对称分布式互斥算法^[2], 但由于该算法的初始化节点数较少, 因此算法的时间复杂度还是比较高。为了改善请求集生成算法的性能, 陈志党在该算法的基础上, 提出了一种通过提高请求集初始化节点的数量的方式来改善请求集生成算法, 即折半循环编码算法^[3], 从而更快、更优地产生请求集, 使算法的时间复杂度大幅度降低。

由于折半循环算法只是简单地提高了算法的时间复杂度和空间复杂度, 算法没有对请求集的长度有所提高, 所生成的请求集长度仍然保持在 $\sqrt{2N} \sim 2\sqrt{N}$ 之间, 因此本文在贪心策略的基础上, 依据贪心策略对可纳入节点进行局部求最优的方式, 来生成请求集的算法, 从而更快、更优地产生请求集。

1 系统模型

设系统的节点数为 N , 并从 $0 \sim N-1$ 对节点编号, 第

* 基金项目: 国家自然科学基金(61063004/F020104); 内蒙自然基金(2010ms0912)

i 个节点的 ID 号为 $i-1$ 。假定系统的节点与通信均可靠, 各节点没有共享存储器和共同的物理时钟, 节点间依靠消息进行异步通信, 并且无法预知消息通信时间延迟。

1.1 对称请求集产生的条件

用 S_N 表示包含 N 个节点的分布式系统, S_i 表示系统中 ID 号为 i 的节点, R_i 表示节点 S_i 的请求集, k, n 等为常数。

MAEKAWA M 提出了对称请求集应满足的四个条件, 即:

A1: $\forall i, j \in [0, N-1], R_i \cap R_j \neq \emptyset$ 。即任意两个节点的请求集交集不为空。

A2: $\forall i \in [0, N-1], S_i \in R_i$ 。即任意节点的请求集包含该节点本身。

A3: $\forall i, j \in [0, N-1], i \neq j, |R_i| = |R_j| = k$ 。即每个节点的请求集长度相同, 都包含 k 个节点。

A4: $\forall i \in [0, N-1], |\{R_j | S_i \in R_j, j \in [0, N-1]\}| = k$, 即任一节点都属于 k 个请求集。

满足条件 A1~A4 的请求集称为对称请求集, 能够生成对称请求集的算法称为对称请求集生成算法, 利用对称请求集实现分布式互斥的算法称为对称分布式互

斥算法。

1.2 请求集产生算法的相关概念

为了减少在生成请求集过程中的循环次数,本文提出了松弛循环差集的定义、贪心算法定义以及循环请求集与松弛差集等价的定理。

定义(贪心策略):贪心策略是指从问题的初始状态出发,通过若干次的贪心选择而得出最优值(或较优解)的一种解题方法。

定理 循环请求集与松弛差集等价。

在循环编码算法中已经证明,循环编码所产生的请求集满足 MAEKAWA M 所提出的四个条件,其产生的请求集是对称请求集,而松弛差集算法中证明了循环请求集与松弛差集等价的定理,因此,松弛循环差集所产生的请求集也是对称请求集。而本算法是在松弛差集算法的基础上进行的改进,即通过增加初始化请求集的长度来缩短算法的时间复杂度,以求更快地找到所求请求集,因此,本算法所产生的请求集也是对称请求集。

1.3 贪心策略理论

1.3.1 贪心选择性质

贪心选择性质是指应用同一规则 f ,将原问题变为一个相似的、但规模更小的子问题,此后的每一步都是当前看似最佳的选择。这种选择依赖于已做出的选择,但不依赖于未做出的选择。从全局来看,运用贪心策略解决的问题在程序的运行过程中无回溯过程。

1.3.2 局部最优解

贪心策略通常是自顶向下进行的。第一步为一个贪心选择,将原问题变成一个相似的、但规模更小的问题,而后的每一步都是当前看似最佳的选择。这种选择可能依赖于已作出的所有选择,但不依赖有待于做的选择或子问题的解。

从求解的全过程来看,每一次贪心选择都将当前问题归纳为更小的相似子问题,而每一个选择都仅做一次,无重复回溯过程。因此,贪心法有较高的时间效率。

2 请求集产生的算法的描述与实现

2.1 数据结构

设系统的节点数为 N ,系统请求集方阵 A_N 是 $N \times N$ 的方阵,其行列编号均为 $0 \sim N-1$, A_N 第 i 行表示系统的第 i 个节点的请求集码字,用 a_i 表示, a_j 表示 A_N 的第 j 行第 i 列元素,它的值表示节点 j 在第 i 个节点的请求集码字中是否被选中, a_{ij} 为 1 表示选中,为 0 表示没被选中。

为了判断是否产生请求集,引进标记数组 T_N ,它具有 N 个分量,每个分量对应 A_N 的一行,该分量为 1 表示 A_N 对应行和第 0 行有交点($(a_0 \& a_i) \neq 0$),反之,则说明没有交点。

表示状态数组 T :长度为 N 的数组。 $T[i]=1$ 用来表示差集 i 在请求集中已被表示, $T[i]=0$ 表示差集 i 在请求集中没被表示。

最小重复记录字 $count$:用来表示第 j 行($1 \leq j < \lceil N/2 \rceil$) 和第 0 行第一次出现没有交点。

2.2 请求集生成算法描述

(1)令 $2^k \leq \lfloor N/2 \rfloor$, 求出 k , 将系统第 0 个节点的码字 a_0 中 $2^0-1, 2^1-1, \dots, 2^k-1$ ($1 \leq k < N$) 位初始化为 1, 并令 $T[0]=1$ 。

(2)对第 0 行向右进行循环右移 i 位($0 \leq i < \lceil N/2 \rceil$) 得到第 i 行, 判断第 0 行和第 i 行是否有交点, 如果有交点, 令 $T[i]=1, i++$, 转(2), 否则转(3)。

(3)若第 0 行和第 i 行没有交点, 则 $count=i$, 遍历第 0 行中 $a[0][j]=0$ ($j \leq j < N$) 的所有情况, 在第 0 行中置 $a[0][j]=1$, 并令 $T[i]=1$, 求以上各种情况下 $count$ 的最大值, 并在 $count$ 取最大值时, 在第一行所对应的节点 j 处, 令 $a[0][j]=1$ 为插入节点, 转(2)。

(4)直到 T 前 $\lceil N/2 \rceil$ 行都为 1 或者 $count > \lceil N/2 \rceil$ 时, 算法结束。

2.3 请求集产生算法的实例实现

以请求集个数 $N=13$ 为例,图 1~图 4 描述了各节点请求集的求取过程。

由于 $2^2 < \lfloor 13/2 \rfloor < 2^3$, 所以 $a[0][0]=1, a[0][1]=1, a[0][3]=1, T[0]=1$ 。

```

1 1 0 1 0 0 0 0 0 0 0 0 0 1
0 1 1 0 1 0 0 0 0 0 0 0 0 1
0 0 1 1 0 1 0 0 0 0 0 0 0 1
0 0 0 1 1 0 1 0 0 0 0 0 0 1
0 0 0 0 1 1 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

图 1 初始化

```

1 1 1 1 0 0 0 0 0 0 0 0 0 1
0 1 1 1 1 0 0 0 0 0 0 0 0 1
0 0 1 1 1 1 0 0 0 0 0 0 0 1
0 0 0 1 1 1 1 0 0 0 0 0 0 1
0 0 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

图 2 第一步令 $a[0][2]=1, count=4$

```

1 1 0 1 1 0 0 0 0 0 0 0 0 1
0 1 1 0 1 1 0 0 0 0 0 0 0 1
0 0 1 1 0 1 1 0 0 0 0 0 0 1
0 0 0 1 1 0 1 1 0 0 0 0 0 1
0 0 0 0 1 1 1 1 0 0 0 0 0 1
0 0 0 0 0 1 1 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

图 3 第三步令 $a[5][4]=1, count=5$

```

1 1 0 1 0 0 0 0 0 0 1 0 0 1
0 1 1 0 1 0 0 0 0 0 1 0 0 1
0 0 1 1 0 1 0 0 0 0 0 1 0 1
0 0 0 1 1 0 1 0 0 0 0 0 1 1
1 0 0 0 1 1 0 1 0 0 0 0 0 1
0 1 0 0 0 1 1 0 1 0 0 0 0 1
0 0 1 0 0 0 1 1 0 1 0 0 0 0

```

图 4 第六步 算法结束

网络与通信 Network and Communication

首先初始化,利用循环右移的方法,当到第四行的时候和第一行没有节点,如图1,则遍历第0行令 $a[0][2]=0$,并令 $T[2]=1$,继续利用循环右移,得到 $count=4$,如图2所示。依次类推,当 $a[0][9]=1$ 时, $count=7$,由于前 $T \lceil N/2 \rceil$ 行都为1,如图4所示,则算法结束。

从以上的计算过程可知,当节点数为13时,本算法的请求集长度与 \sqrt{N} 长度相同,请求集为[110100000100000000]。而在折半循环编码算法中,产生的请求集长度为5。

3 性能分析

分布式互斥请求集生成算法的性能度量主要有三个指标:请求集长度、时间复杂度和空间复杂度。本文将分别分析这些指标。

3.1 请求集长度说的

表1比较了几种典型分布式互斥算法所产生的请求集长度,其中 Bin-cyclic 代表折半循环编码算法。

表1 几种分布式互斥算法的请求集大小的比较

N	13	31	57	91	133	241	343	381	507	1 000	5 000	10 000	20 000
$\sqrt{2N}$	5	8	11	13	16	22	26	28	32	45	100	142	200
\sqrt{N}	4	5	8	10	12	16	19	20	23	32	71	100	142
Bin-cyclic	5	8	11	14	19	28	36	40	43	72	161	244	359
本算法	4	5	9	11	12	17	19	21	23	33	73	107	145

由表1可知,本算法产生的请求集大小与 \sqrt{N} 接近,所生成的请求集长度比折半循环编码互斥算法生成的请求集长度小,更趋于 \sqrt{N} ,接近 MAEKAWA M 算法所能达到的数量级。折半循环编码算法所得到的请求集长度在节点数 $N \leq 1000$ 时,可以保持在 $\sqrt{2N} \sim 2\sqrt{N}$ 之间,但当节点数 $N \geq 2000$ 时,其所产生请求集的长度就超出了 $2\sqrt{N}$,而且随着节点数 N 的增大,所得请求集的长度就超出越多;而本算法所得请求集的长度基本与 \sqrt{N} 相近,即使当节点数 N 达到 20000 时,其所得请求集的长度仍然接近 \sqrt{N} 。

3.2 时间复杂度

折半循环编码算法的时间复杂度为 $2\sqrt{N} - \log_2 \lfloor N/$

$2 \rfloor - 1$,本算法的请求集数量级为 \sqrt{N} ,故时间复杂度为 $\sqrt{N} \times (\sqrt{N} - 1)/2 \approx N/2$,因此本算法的时间复杂度为 $O(N/2)$,比如当 $N=100$ 时,本算法的时间复杂度为 45。

3.3 空间复杂度

由于折半循环算法引入了对称请求集的概念,只计算前 $\lceil N/2 \rceil$ 行和第0行有交点即可。空间复杂度为 $O(N^2/2)$,而本算法只是在贪心策略的基础上,依据贪心策略对可纳入节点通过局部求最优的方式来生成请求集,所以,空间复杂度与折半循环算法一样。

公平、健壮和易于实现的分布式互斥算法是研究人员追求的最终目标,本文在陈志党提出的折半循环编码算法的基础上,依据贪心策略对可纳入节点进行局部求最优的方式来生成请求集的算法,从而产生更优的请求集,使算法的请求集长度下降到 \sqrt{N} ,空间复杂度可近似为 $O(N^2/2)$,时间复杂度为 $O(N/2)$,因此本算法易于实际应用。

参考文献

- [1] MAEKAWA M. A Nalgorithm for mutual exclusion in decentralized systems [J]. ACM Transactions on Computer Systems, 1985, 3(2):145-159.
- [2] 李美安,刘心松,王征.一种基于松弛循环差集的高性能分布式互斥算法[J].电子学报,2007,35(1):58-63.
- [3] 陈志党,李美安.一种新的分布式互斥请求集生成算法[J].微计算机信息,2010(3-9):211-212.
- [4] 申二威,李美安.一种改进的高效分布式互斥请求集生成算法[J].微计算机信息,2010(8-24):201-20.

(收稿日期:2010-12-10)

作者简介:

李美安,男,1973年生,教授,硕士生导师,主要研究方向:分布式计算,分布式操作系统,宽带网络与通信等。

陈志党,男,1985年生,硕士研究生,主要研究方向:分布式计算,分布式操作系统,宽带网络与通信等。

王春申,女,1985年生,硕士研究生,主要研究方向:图像处理,图像复原等。